

1 **24.** (twice amended) An improved server of the type that provides a program with a standard
2 interface for performing queries on querying-remote datasets, a query being able to specify a
3 subset of the dataset in terms of values in the dataset, and

4 the improved server comprising:

5 a queryable cache that contains copies of certain of the datasets and is local to the server,
6 the improved server being configured to receive-receiving a query for-on a remote dataset in a
7 form required by the interface from the program, determining whether a copy of a-the remote
8 dataset to-be-queried is present in the queryable cache, and, if the copy is present, performing
9 the query on querying-the copy, and otherwise-queryingperforming the query on the remote
10 dataset,

11 whereby the queryable cache is transparent to the program.

FAX RECEIVED

APR 01 2003

GROUP 3600
OFFICIAL

1 **25.** (amended) The improved server set forth in claim 24 wherein

2 the program uses global identifiers for the remote data sets and

3 the copies in the queryable cache have local identifiers; and

4 the improved server further comprises:

5 a query analyzer that receives the global identifier for a dataset being queried and if
6 there is a copy of the data set indicated by the global identifier, returns the local identifier to the
7 server,

8 the server using the local identifier to query-perform the query on the copy.

1 **26.** (amended) The improved server set forth in claim 25 wherein:

2 the query analyzer further indicates to the server whether the copy of the dataset is in the
3 queryable cache.

1 **27.** (amended) The improved server set forth in claim 24 further comprising:

2 a dataset manager that determines a dataset for which a copy is needed in the cache,
3 obtains a copy of the remote dataset, and adds the copy to the cache.

1 **28.** (amended) The improved server set forth in claim 27 wherein:

2 the dataset manager further determines a dataset for which a copy is no longer needed in
3 the cache and removes the copy from the cache.

1 **29.** (amended) The improved server set forth in any of claims 27 or 28 wherein:

2 the dataset manager determines whether to add or remove a dataset by determining a
3 likelihood that a query will be ~~made to~~performed on the dataset.

1 **30.** (amended) The improved server set forth in claim 29 wherein the improved server further
2 comprises:

3 a query log that lists past queries that have been made to the standard interface and

4 the dataset manager uses the query log to determine a likelihood that a query will be
5 ~~made to~~performed on a dataset.

1 **31.** (amended) The improved server set forth in claim 29 wherein:

the dataset manager uses information about an event that will result in queries to a dataset to determine a likelihood that a query will be ~~made to~~performed on a dataset.

32. (amended) The improved server set forth in claim 31 wherein:

the dataset manager uses information about a time of occurrence of the event to determine a likelihood that a query will be ~~made to~~performed on a dataset.

33. (amended) The improved server set forth in claim 24 wherein

when a change occurs in a remote dataset of the remote datasets, an indication including the change is sent to the server without intervention by the server and

the improved server further comprises:

an update receiver that receives the indication and modifies any copy of the changed dataset as required by the indication.

34. (amended) A method of ~~querying for~~performing queries on datasets in a server that

provides a standard interface for ~~querying performing queries on~~ remote data sets to a program executing on the server, a query being able to specify a subset of the dataset in terms of values in the dataset and

the method comprising the steps of:

receiving a query for a remote dataset in a form required by the standard interface;

determining whether a copy of ~~a the remote~~ dataset to be queried is present in a

queryable cache local to the server; and

9 if the copy is present in the queryable cache, ~~querying-performing the query on the~~
10 copy and otherwise ~~querying-performing the query on the~~ remote dataset,
11 whereby the queryable cache is transparent to the program.

1 35. The method set forth in claim 34 wherein

2 the form required by the standard interface uses global identifiers for the remote data
3 sets and

4 the copies in the queryable cache have local identifiers; and

5 the method further includes the steps of :

6 providing the global identifier for a dataset being queried to a query analyzer in the
7 server; and

8 if there is a copy of the data set indicated by the global identifier, receiving the local
9 identifier from the query analyzer,

10 the local identifier being used in the step of ~~querying-performing the query on the local~~
11 copy.

1 53. The method set forth in claim 35 further comprising the step of:

2 receiving an indication from the query analyzer whether the copy is present in the
3 queryable cache.

1 54. The method set forth in claim 34 further comprising the steps of:

2 determining a dataset for which a copy is needed in the cache;
3 obtaining the copy; and

4 adding the copy to the cache.

1 55. The method set forth in claim 54 further comprising the steps of:

2 determining a dataset for which a copy is no longer needed in the cache; and

3 removing the copy from the cache.

1 56. The method set forth in any one of claims 27 or 28 wherein:

2 the step of determining a dataset is performed by determining a likelihood that a

3 query will be ~~made to~~ performed on the dataset.

1 57. (amended) The method set forth in claim 56 wherein:

2 in the step of determining a dataset, a query log that lists past queries is used to

3 determine the likelihood that a query will be ~~made~~ performed on the dataset.

1 58. The method set forth in claim 56 wherein:

2 in the step of determining a dataset, information about an event that will result in

3 queries to a dataset is used to determine the likelihood that a query will be

4 ~~made~~ performed on the dataset.

1 59. The method set forth in claim 58 wherein:

2 in the step of determining a dataset, information about a time of occurrence of the

3 event is used to determine the likelihood that a query will be ~~made~~ performed on the

4 dataset.

1 **60.** The method set forth in claim 34 wherein

2 when a change occurs in a remote dataset of the remote datasets, an indication

3 including the change is sent to the server without intervention by the server and

4 the method further comprises the steps of:

5 receiving the indication and modifying any copy of the changed dataset as required

6 by the indication.

1 **84.** A memory device characterized in that:

2 the memory device contains code which, when executed by a processor, performs a

3 method of ~~querying~~for performing queries on datasets in a server that provides a standard

4 interface for ~~querying~~performing queries on remote data sets to a program executing on the

5 server, a query being able to specify a subset of the dataset in terms of values in the dataset and

6 the method comprising the steps of

7 receiving a query for a remote dataset in a form required by the standard interface;

8 determining whether a copy of ~~a~~the remote dataset to be queried is present in a

9 queryable cache local to the server; and

10 if the copy is present in the queryable cache, ~~querying~~performing the query on the copy

11 and otherwise ~~querying~~performing the query on the remote dataset,

12 whereby the queryable cache is transparent to the program.

1 **85.** The memory device set forth in claim 84 further characterized in that:

2 in the method, the form required by the standard interface uses global identifiers

3 for the remote data sets and

4 the copies in the queryable cache have local identifiers; and
5 the method further includes the steps of :
6 providing the global identifier for a dataset being queried to a query analyzer in
7 the server; and
8 if there is a copy of the data set indicated by the global identifier, receiving the
9 local identifier from the query analyzer,
10 the local identifier being used in the step of ~~querying~~ performing the query on the local
11 copy.

1
1 **86.** The memory device set forth in claim 84 wherein the method further comprises the
2 steps of:
3 determining a dataset for which a copy is needed in the cache;
4 obtaining the copy; and
5 adding the copy to the cache.

1 **87.** The memory device set forth in claim 86 wherein the method further comprises the
2 steps of:
3 determining a dataset for which a copy is no longer needed in the cache; and
4 removing the copy from the cache.

1 **88.** The memory device set forth in any one of claims 86 or 87 wherein:
2 the step of determining a dataset is performed by determining a likelihood that a
3 query will be ~~made to~~ performed on the dataset.

1 **89.** (amended) The memory device set forth in claim 88 wherein:

2 in the step of determining a dataset, a query log that lists past queries is used to
3 determine the likelihood that a query will be ~~made~~performed on the dataset.

1 **90.** The memory device set forth in claim 88 wherein:

2 in the step of determining a dataset, information about an event that will result in
3 queries to a dataset is used to determine the likelihood that a query will be
4 ~~made~~performed on the dataset.

1 **91.** The memory device set forth in claim 90 wherein:

2 in the step of determining a dataset, information about a time of occurrence of the
3 event is used to determine the likelihood that a query will be ~~made~~performed on the
4 dataset.

1 **92.** The memory device set forth in claim 84 wherein

2 when a change occurs in a remote dataset of the remote datasets, an indication
3 including the change is sent to the server without intervention by the server and
4 the method further comprises the steps of:

5 receiving the indication and modifying any copy of the changed dataset as
6 required by the indication.

1 **93.** The memory device set forth in claim 85 wherein the method further comprises the
2 step of:

3 receiving an indication from the query analyzer whether the copy is present in the
4 queryable cache.

1 **112.** Apparatus for responding to a request, the request including one or more specifiers
2 referring to objects belonging to a plurality thereof in a distributed database system that
3 includes a plurality of database systems and
4 the apparatus comprising:

5 a first database system of the plurality; and

6 a redirector which responds to the request when the request includes a specifier
7 that cannot be interpreted in the first database system by causing the request to be
8 executed at least in part in a second database system of the plurality, the request
9 otherwise being executed in the first database system.

1 **113.** The apparatus set forth in claim 112 wherein:

2 the objects in the first database system include copies of objects contained
3 in at least one other database system belonging to the distributed database system.

1 **114.** The apparatus set forth in claim 113 wherein:

2 the first database system functions as a cache with regard to the objects whose
3 copies are included therein.

1 **115.** The apparatus set forth in claim 113 wherein the other database system is the
2 second database system.

1 **116.** The apparatus set forth in claim 115 wherein:

2 the first database system functions as a cache with regard to the second database

3 system.

1 **117.** The apparatus set forth in any one of claims 112 through 116 wherein:

2 the apparatus is local to a server of the type that provides a program executing in
3 the server with a standard interface for querying databases; and
4 the requests include queries received via the standard interface.

1 **118.** The apparatus set forth in claim 117 wherein:

2 the server obeys the http protocol and the program is a Web application program.

1 **119.** A method of responding to a request, the request including one or more specifiers
2 that refer to objects belonging to a plurality thereof in a distributed database system that
3 includes a plurality of database systems and
4 the method comprising the steps of:

5 receiving the request in a first database system of the plurality;
6 determining whether the request includes a specifier that cannot be interpreted in
7 the first database system of the plurality; and
8 when the request includes such a specifier, causing the request to be executed at
9 least in part in a second database system of the plurality.

1 **120.** The method set forth in claim 119 wherein:

2 the objects in the first database system include copies of objects contained in at
3 least one other database system belonging to the distributed database system,

4 whereby the first database system functions as a cache with regard to the objects whose
5 copies are included therein.

1 **121.** The method set forth in claim 120 wherein:

2 the other database system is the second database system,
3 whereby the first database system functions as a cache with regard to the second database
4 system.

1 **122.** The method set forth in any one of claims 119 through 121 wherein:

2 the first database system is local to a server of the type that provides a program
3 executing in the server with a standard interface for querying databases; and
4 in the step of receiving the request, the request is received via the standard
5 interface.

1 **123.** The method set forth in claim 122 wherein:

2 the server obeys the http protocol and the program is a Web application program.

1 **124.** A memory device characterized in that:

2 the memory device contains code which, when executed in a processor, performs
3 a method of responding to a request, the request including one or more specifiers that
4 refer to objects belonging to a plurality thereof in a distributed database system that
5 includes a plurality of database systems and
6 the method comprising the steps of:

7 receiving the request in a first database system of the plurality;
8 determining whether the request includes a specifier that cannot be interpreted in
9 the first database system of the plurality; and
10 when the request includes such a specifier, causing the request to be executed at
11 least in part in a second database system of the plurality.

1 **125.** Apparatus for caching copies of objects belonging to a subset of the objects
2 belonging to a first database system that returns an object in response to a request
3 therefor, the request including one or more specifiers referring to the objects and
4 the apparatus comprising:

5 a second database system that contains the copies; and
6 a redirector that responds to the request when the request includes a specifier that
7 cannot be interpreted in the second database system by causing the request to be executed
8 at least in part in the first database system, the request otherwise being executed in the
9 second database system.

1 **126.** The apparatus set forth in claim 125 wherein:

2 the apparatus is local to a server of the type that provides a program executing in
3 the server with a standard interface for querying databases; and
4 the requests include queries received via the standard interface.

1 **127.** The apparatus set forth in claim 126 wherein:

2 the server obeys the http protocol and the program is a Web application program.

1 **128.** A method of responding to a request that includes one or more specifiers referring to
2 objects belonging to a set thereof where the objects are stored in a first database system
3 and copies of a subset thereof are stored in a second database system,

4 the method comprising the steps of:

5 receiving the request in the second database system;

6 determining whether the request includes a specifier that cannot be interpreted in the
7 second database system; and

8 when the request includes such a specifier, causing the request to be executed at least in
9 part in the first database system instead of in the second database system.

1 **129.** The method set forth in claim 128 wherein:

2 the second database system is local to a server of the type that provides a program
3 executing in the server with a standard interface for querying databases; and

4 in the step of receiving the request, the request is received via the standard interface.

1 **130.** The method set forth in claim 129 wherein:

2 the server obeys the http protocol and the program is a Web application program.

1 **131.** A memory device characterized in that:

2 the memory device contains code which, when executed in a processor, performs
3 a method of responding to a request that includes one or more specifiers referring to
4 objects belonging to a set thereof where the objects are stored in a first database system

5 and copies of a subset thereof are stored in a second database system,
6 the method comprising the steps of:
7 receiving the request in the second database system;
8 determining whether the request includes a specifier that cannot be interpreted in
9 the second database system; and
10 when the request includes such a specifier, causing the request to be executed at
11 least in part in the first database system instead of in the second database system.

1